

目录

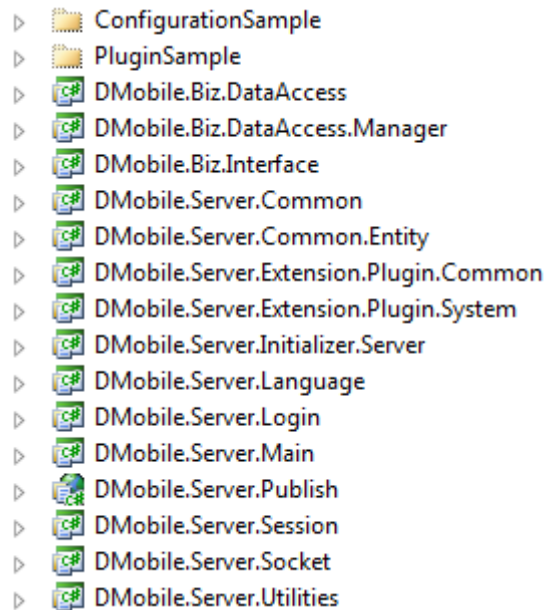
- 简介 .....1
  - 目的.....1
  - 框架文件结构.....2
  - 业务文件结构.....2
- 部署 .....3
  - 配置文件.....3
    - 本地配置.....3
    - IIS 配置.....5
  - 服务寄宿.....6
    - 本地宿主/Windows 服务 .....6
    - IIS 寄宿.....6
    - 文件结构.....7
- 框架实现.....7
  - 流程图.....7
  - 客户端请求数据格式.....8

简介

目的

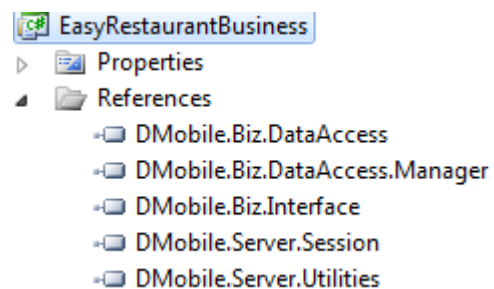
DMobile.Framework 目的是为了更方便 iOS、Android、Win8 等移动客户端与服务器交互的轻量级服务器框架。

## 框架文件结构



- 1、DMobile.Biz 为一系列业务接口和数据库方法，开发人员使用的业务必须继承 DMobile.Biz.Interface，数据库方法则不是必须的。
- 2、DMobile.Server 为服务器通讯、扩展插件、发布、会话等方法，开发人员扩展插件、指定服务器通讯内容解析方法应继承相应接口实现。
- 3、Sample 展示了如何继承并实现。

## 业务文件结构

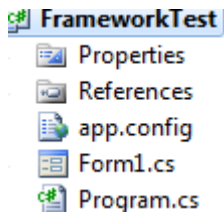


这是一个业务示例，业务应继承 DMobile.Biz.Interface

# 部署

## 配置文件

## 本地配置



配置程序配置文件，加入服务器节点，和服务器配置

```
<configuration>
<startup>
  <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/>
</startup>
<system.web>
  <httpRuntime maxRequestLength="65535"/>
</system.web>
<system.serviceModel>
  <standardEndpoints>
    <webHttpEndpoint>
      <standardEndpoint name="LoginEndPoint" maxBufferSize="2147483647"
        maxBufferPoolSize="2147483647"
maxReceivedMessageSize="2147483647"
        defaultOutgoingResponseFormat="Json" helpEnabled="true">
        <readerQuotas maxDepth="2147483647"
maxStringContentLength="2147483647"
          maxArrayLength="2147483647" maxBytesPerRead="2147483647"
maxNameTableCharCount="2147483647" />
      </standardEndpoint>
    </webHttpEndpoint>
  </standardEndpoints>
  <behaviors>
    <serviceBehaviors>
      <behavior name="LoginServiceBehavior">
        <bufferedReceive maxPendingMessagesPerChannel="1024" />
        <serviceThrottling />
      </behavior>
    </serviceBehaviors>
  </behaviors>
</system.serviceModel>
</configuration>
```

```

    <bindings>
      <webHttpBinding>
        <binding name="LoginBinding" maxBufferSize="2147483647"
maxBufferPoolSize="2147483647"
          maxReceivedMessageSize="2147483647">
          <readerQuotas maxDepth="2147483647"
maxStringContentLength="2147483647"
            maxArrayLength="2147483647" maxBytesPerRead="2147483647"
maxNameTableCharCount="2147483647" />
        </binding>
        <binding name="CommunicationBinding" maxBufferSize="2147483647"
          maxBufferPoolSize="2147483647"
maxReceivedMessageSize="2147483647">
          <readerQuotas maxDepth="2147483647"
maxStringContentLength="2147483647"
            maxArrayLength="2147483647" maxBytesPerRead="2147483647"
maxNameTableCharCount="2147483647" />
        </binding>
      </webHttpBinding>
    </bindings>
    <services>
      <service name="DMobile.Server.Main.CommunicationCenter">
        <endpoint address="http://localhost:4477/CommunicationCenter"
          binding="webHttpBinding"
bindingConfiguration="CommunicationBinding"
          name="basicWeb"
contract="DMobile.Server.Main.ICommunicationCenter" />
      </service>
      <service name="DMobile.Server.Login.LoginService">
        <endpoint address="http://localhost:4477/LoginService"
          binding="webHttpBinding"
          bindingConfiguration="LoginBinding" name="Login"
contract="DMobile.Server.Login.ILoginService"
          kind="webHttpEndpoint" endpointConfiguration="LoginEndPoint" />
      </service>
    </services>
  </system.serviceModel>
  <appSettings>
    <!--Configer-->
    <add key="ServerConfigFile" value="DMobile.Server.Configuration.dll"/>
    <add key="BizFile" value="EasyRestaurantBusiness.dll"/>
  </appSettings>
</configuration>

```

- 1、注意 appSettings 的节点路径，其中 BizFile 不是必须的，但在 bin 文件夹业务文件下

- 应以 Business 结尾。ServerConfigFile 必须指定服务器配置方法的 dll。
- 2、Endpoint 节点地址可以任意修改，不过要符合地址规则。

## IIS 配置

<configuration>

<system.web>

<compilation debug="true" targetFramework="4.0" />

<httpRuntime maxRequestLength="65535" />

</system.web>

<system.webServer>

<modules runAllManagedModulesForAllRequests="true">

<add name="UrlRoutingModule"

type="System.Web.Routing.UrlRoutingModule, System.Web, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />

</modules>

</system.webServer>

<system.serviceModel>

<serviceHostingEnvironment aspNetCompatibilityEnabled="true" />

<standardEndpoints>

<webHttpEndpoint>

<!--

Configure the WCF REST service base address via the global.asax.cs file and the default endpoint

via the attributes on the <standardEndpoint> element below

-->

<standardEndpoint name="" helpEnabled="true"

automaticFormatSelectionEnabled="true" />

</webHttpEndpoint>

</standardEndpoints>

</system.serviceModel>

<appSettings>

<!--IIS Configer-->

<add key="ServerConfigFile" value="bin\DMobile.Server.Configuration.dll"/>

<add key="BizFile" value="bin\EasyRestaurantBusiness.dll"/>

</appSettings>

</configuration>

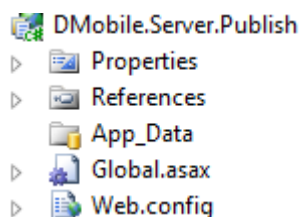
## 服务寄宿

### 本地宿主/Windows 服务

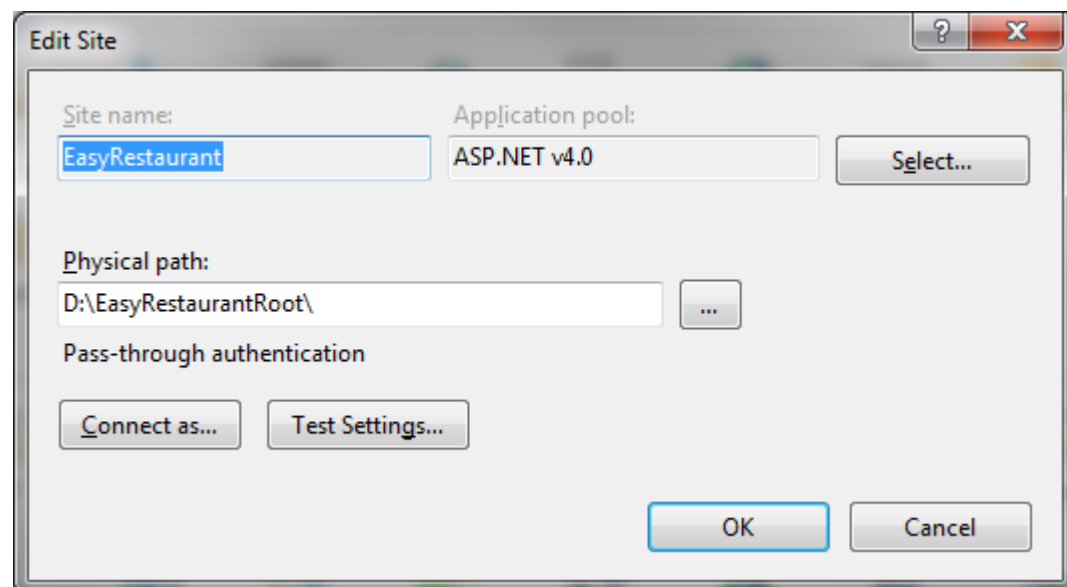
```
private void OpenLogin()
{
    using (WebServiceHost login = new WebServiceHost(typeof(LoginService)))
    {
        login.Open();
        while (manualEvent.WaitOne())
        {
            break;
        }
    }
}
```

在程序或者服务中，使用以上代码就可以开启服务了。  
其中 manualEvent 目的是不让这个线程退出，保持服务运行。

## IIS 寄宿



- 1、右键单击项目，选择发布（publish）
- 2、选择文件系统，发布至 IIS 指定的目录



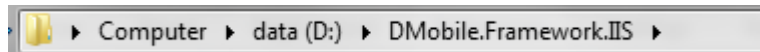
例如：我的 IIS 指定为 D:\EasyRestaurantRoot\,则发布至这个文件夹下，就可以访问了。  
细节请看 <http://www.cnblogs.com/KeithWang/archive/2012/02/14/2351826.html>

## 文件结构

- 1、保证 plugin 在 plugins 文件夹下
- 2、保证业务文件在 bin 下

以下为一个例子：

第一级：



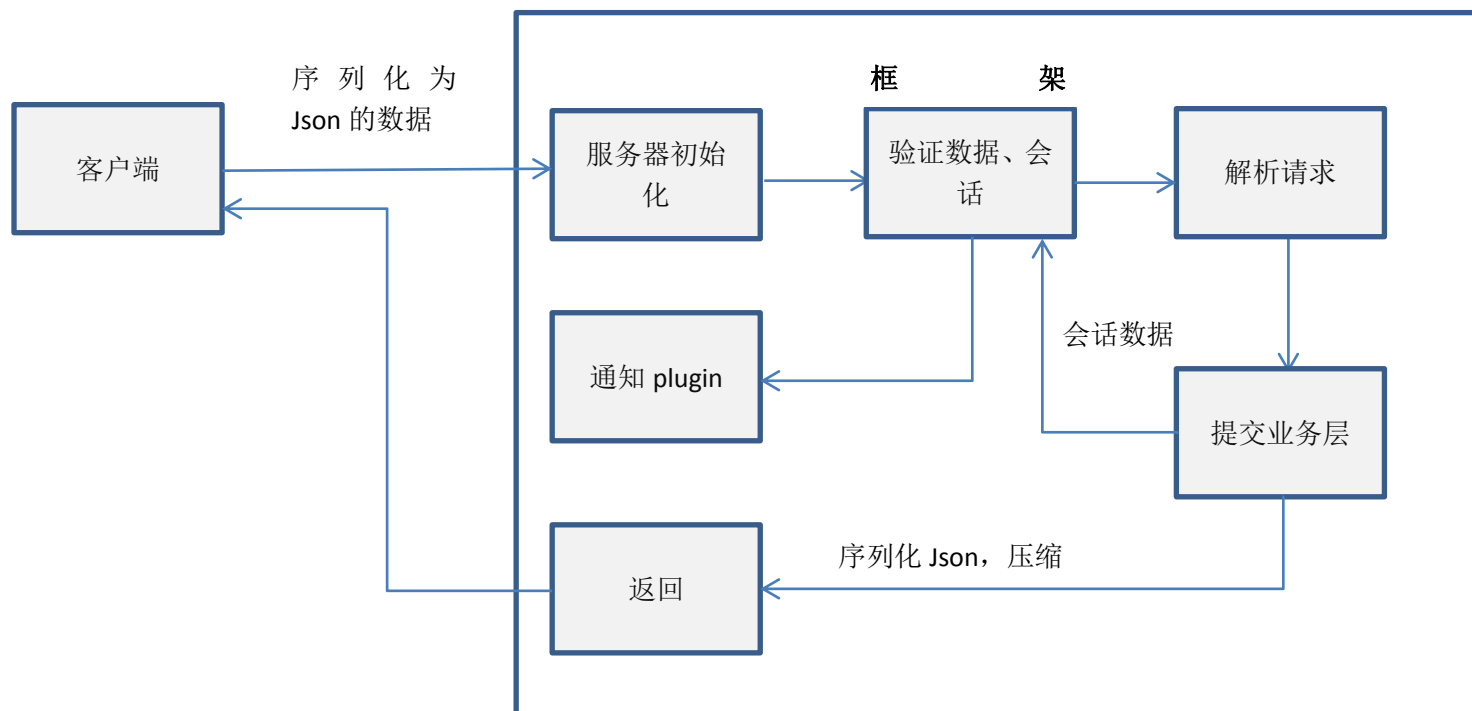
Name	Date modified	Type	Size
bin	2013/3/9 13:13	File folder	
Global.asax	2013/3/9 9:55	ASP.NET Server A...	1 KB
Web.config	2013/3/9 13:13	XML Configuratio...	2 KB

Bin:

Name	Date modified	Type	Size
zh-CHS	2013/3/9 13:13	File folder	
DMobile.Biz.Interface.dll	2013/3/9 13:13	Application extens...	5 KB
DMobile.Server.Common.dll	2013/3/9 13:13	Application extens...	16 KB
DMobile.Server.Common.Entity.dll	2013/3/9 13:13	Application extens...	6 KB
DMobile.Server.Configuration.dll	2013/3/9 13:13	Application extens...	10 KB
DMobile.Server.Extension.Plugin.Commo...	2013/3/9 13:13	Application extens...	6 KB
DMobile.Server.Extension.Plugin.System....	2013/3/9 13:13	Application extens...	9 KB
DMobile.Server.Initializer.Server.dll	2013/3/9 13:13	Application extens...	8 KB
DMobile.Server.Language.dll	2013/3/9 13:13	Application extens...	6 KB
DMobile.Server.Login.dll	2013/3/9 13:13	Application extens...	9 KB
DMobile.Server.Main.dll	2013/3/9 13:13	Application extens...	9 KB
DMobile.Server.Publish.dll	2013/3/9 13:13	Application extens...	5 KB
DMobile.Server.Session.dll	2013/3/9 13:13	Application extens...	14 KB
DMobile.Server.Socket.dll	2013/3/9 13:13	Application extens...	4 KB
DMobile.Server.Utilities.dll	2013/3/9 13:13	Application extens...	20 KB
Newtonsoft.Json.dll	2013/3/9 8:29	Application extens...	383 KB

## 框架实现

### 流程图



## 客户端请求数据格式

格式统一为 Json 格式。

请求数据格式分为两部分：

每部分格式都为{"Key":key,"Value":value}

1、所请求的方法{"Key":"METHOD","Value":"方法格式 Json"}

2、所附加的信息{"Key":"ATTACHMENT","Value":"附加信息 Json"}

方法格式与 DMobile.Server.Common.MethodSchema 一致，附加信息与 DMobile.Server.Common.AttachmentSchema 一致

例如：

```
[{"Key":"METHOD","Value":{"MethodName":"OperationUserLocation","Parameters":{"location":{"UserID":"admin","IPAddress":"127.0.0.1","Port":0,"NetWorkType":0,"Latitude":33.2355,"Longitude":-325.232,"City":30,"OnlineStatus":0},"operationType":"2"}}},{Key":"ATTACHMENT","Value":{"MacAddress":"Empty"}}]
```

## 服务器返回数据格式

格式统一为 Json 格式（未压缩情况下）

例如：

```
{"location":{"UserID":"admin","IPAddress":"127.0.0.1","Port":0,"NetWorkType":0,"Latitude":33.2355,"Longitude":-325.232,"City":30,"OnlineStatus":0}}
```



# 服务器流程

## 初始化

在服务端（LoginService、CommunicationCenter）构造函数中

```
PluginManager.Instance.LoadPlugins(PluginDetector.GetPlugins());
```

```
MethodRoute.Register(ConfigurationLoader.Instance.Business.GetType());
```

这两个函数分别加载插件和业务

## URI

### LoginServic

Get 方法: `http://节点地址/request?data={encryptData}`

POST 方法: `http://节点地址/post/{dataLength}`, 传入的流数据为 Json 数据

### CommunicationCenter

Get 方法: `http://节点地址/request?sessionId={sessionId}&encryptdata={encryptData}`

POST 方法: `http://节点地址/post/{dataLength}`, 传入的流数据为 Json 数据

## 数据解析

服务器将使用 **IAttachmentResolver**, **IDataParser**, **IMethodResolver**, **IRequestSplitter**, **ISecurityAuthorize** 这几个接口对传入的数据进行解析, 具体请看 `DMobile.Server.Configuration`, 服务器加载此配置作为解析器。

**IDataParser**: 数据加密解密, 请保持客户端、服务端使用同一种算法加解密

**IMethodResolver**: 将对请求的方法解析, 以适配业务层的方法 (NOTE: 一般不重新定义此接口的实现)

**IRequestSplitter**: 将对客户端传入的数据解析成各个子字符串以备对应解析方法使用

**ISecurityAuthorize**: 扩展方法, 安全认证的解析

**IAttachmentResolver**: 附加信息解析

# 版本

## 版本号

### **1.0.0.0**

作者：Jason(董健) 成都

日期：2013 年 3 月 9 日

### **1.0.0.1**